Truthful Mechanism Design for Multidimensional Covering Problems

Hadi Minooei* Chaitanya Swamy*

Abstract

We investigate *multidimensional covering mechanism-design* problems, wherein there are m items that need to be covered and n agents who provide covering objects, with each agent i having a private cost for the covering objects he provides. The goal is to select a set of covering objects of minimum total cost that together cover all the items.

We focus on two representative covering problems: uncapacitated facility location (UFL) and vertex cover (VC). For multidimensional UFL, we give a black-box method to transform any *Lagrangian-multiplier-preserving* ρ-approximation algorithm for UFL to a truthful-in-expectation, ρ-approx. mechanism. This yields the first result for multidimensional UFL, namely a truthful-in-expectation 2-approximation mechanism.

For multidimensional VC (Multi-VC), we develop a *decomposition method* that reduces the mechanism-design problem into the simpler task of constructing *threshold mechanisms*, which are a restricted class of truthful mechanisms, for simpler (in terms of graph structure or problem dimension) instances of Multi-VC. By suitably designing the decomposition and the threshold mechanisms it uses as building blocks, we obtain truthful mechanisms with approximation ratios (n is the number of nodes): (1) $O(\log n)$ for Multi-VC on any proper minor-closed family of graphs; and (2) $O(r^2 \log n)$ for r-dimensional VC on any graph. These are the first truthful mechanisms for Multi-VC with non-trivial approximation guarantees.

1 Introduction

Algorithmic mechanism design (AMD) deals with efficiently-computable algorithmic constructions in the presence of strategic players who hold the inputs to the problem, and may misreport their input if doing so benefits them. The challenge is to design algorithms that work well with the true (privately-known) input. In order to achieve this task, a *mechanism* specifies both an algorithm and a pricing or payment scheme that can be used to incentivize players to reveal their true inputs. A mechanism is said to be *truthful*, if each player maximizes his utility by revealing his true input regardless of the other players' declarations.

In this paper, we initiate a study of *multidimensional covering mechanism-design* problems, often called *reverse auctions* or *procurement auctions* in the mechanism-design literature. These can be abstractly stated as follows. There are m items that need to be covered and n agents who provide covering objects, with each agent i having a private cost for the covering objects he provides. The goal is to select (or buy) a suitable set of covering objects from each player so that their union covers all the items, and the total covering cost incurred is minimized. This *cost-minimization* (CM) problem is equivalent to the *social-welfare maximization* (SWM) (where the social welfare is $-$ (total cost incurred by the players and the mechanism designer)), so ignoring computational efficiency, the classical VCG mechanism [27, 4, 15] yields a truthful mechanism that always returns

*{hminooei,cswamy}@math.uwaterloo.ca. Dept. of Combinatorics and Optimization, Univ. Waterloo, Waterloo, ON N2L 3G1. Research supported partially by NSERC grant 327620-09 and the second author's Ontario Early Researcher Award.

an optimal solution. However, the CM problem is often *NP*-hard, so we seek to design a *polytime* truthful mechanism where the underlying algorithm returns a near-optimal solution to the CM problem.

Although multidimensional packing mechanism-design problems have received much attention in the AMD literature, multidimensional covering CM problems are conspicuous by their absence in the literature. For example, the packing SWM problem of combinatorial auctions has been studied (in various flavors) in numerous works both from the viewpoint of designing polytime truthful, approximation mechanisms [10, 21, 9, 13], and from the perspective of proving lower bounds on the capabilities of computationally- (or query-) efficient truthful mechanisms [20, 14, 11]. In contrast, the lack of study of multidimensional covering CM problems is aptly summarized by the blank table entry for results on truthful approximations for procurement auctions in Fig. 11.2 in [26] (a recent result of [12] is an exception; see "Related work"). In fact, to our knowledge, the only multidimensional problem with a covering flavor that has been studied in the AMD literature is the makespan-minimization problem on unrelated machines [25, 22, 2], which is not an SWM problem.

Our results and techniques. We study two representative multidimensional covering problems, namely (metric) *uncapacitated facility location (UFL)*, and *vertex cover* (VC), and develop various techniques to devise polytime, truthful, approximation mechanisms for these problems.

For multidimensional UFL (Section 3), wherein players own (known) different facility sets and the assignment costs are public, we present a *black-box reduction from truthful mechanism design to algorithm design.* We show that any ρ-approximation algorithm for UFL satisfying an additional *Lagrangian-multiplier-preserving* (LMP) property (that indeed holds for various algorithms) can be converted in a black-box fashion to a truthful-in-expectation ρ-approximation mechanism (Theorem 3.1). This is the *first* such black-box reduction for a multidimensional covering problem, and it leads to the first result for multidimensional UFL, namely, a truthful-in-expectation, 2-approximation mechanism. Our result builds upon the convex-decomposition technique in [21]. Lavi and Swamy [21] primarily focus on packing problems, but remark that their convex-decomposition idea also yields results for *single-dimensional* covering problems, and leave open the problem of obtaining results for multidimensional covering problems. Our result for UFL identifies an interesting property under which a ρ-approximation algorithm for a covering problem can be transformed into a truthful, ρ-approximation mechanism in the multidimensional setting.

In Section 4, we consider multidimensional VC, where each player owns a (known) set of nodes. Although, algorithmically, VC is one of the simplest covering problems, it becomes a surprisingly challenging mechanism-design problem in the *multidimensional* mechanism-design setting, and, in fact, seems significantly more difficult than multidimensional UFL. This is in stark contrast with the single-dimensional setting, where each player owns a single node. Before detailing our results and techniques, we mention some of the difficulties encountered. We use Multi-VC to distinguish the multidimensional mechanism-design problem from the algorithmic problem.

For *single-dimensional* problems, a simple monotonicity condition characterizes the *implementability* of an algorithm, that is, whether it can be combined with suitable payments to obtain a truthful mechanism. This condition allows for ample flexibility and various algorithm-design techniques can be leveraged to design monotone algorithms for both covering and packing problems (see, e.g., [3, 21]). For single-dimensional VC, many of the known 2-approximation algorithms for the algorithmic problem (based on LP-rounding, primal-dual methods, or combinatorial methods) are either already monotone, or can be modified in simple ways so that they become monotone, and thereby yield truthful 2-approximation mechanisms [7]. However, the underlying algorithm-design techniques fail to yield algorithms satisfying *weak monotonicity* (WMON)—a necessary condition for implementability (see Theorem 2)—even for the simplest multidimensional setting, namely, 2-

dimensional VC, where *every player owns at most two nodes*. We show this for various LP-rounding methods in Appendix B, and for primal-dual algorithms in Appendix C.

Furthermore, various techniques that have been devised for designing polytime truthful mechanisms for multidimensional packing problems (such as combinatorial auctions) do not seem to be helpful for Multi-VC. For instance, the well-known technique of constructing a *maximal-in-range*, or more generally, a *maximal-in-distributional-range* (MIDR) mechanism—fix some subset of outcomes and return the best outcome in this set—does not work for Multi-VC [12] (and more generally, for multidimensional covering problems). (More precisely, any algorithm for Multi-VC whose range is a proper subset of the collection of minimal vertex covers, cannot have bounded approximation ratio.) This also rules out the convex-decomposition technique of [21], which we exploit for multidimensional UFL, because, as noted in [21], this yields an MIDR mechanism.

Thus, we need to develop new techniques to attack Multi-VC (and multidimensional covering problems in general). We devise two main techniques for Multi-VC. We introduce a simple class of truthful mechanisms called *threshold mechanisms* (Section 4.1), and show that despite their restrictions, threshold mechanisms can achieve non-trivial approximation guarantees. We next develop a *decomposition method* for Multi-VC (Section 4.2) that provides a general way of reducing the mechanism-design problem for Multi-VC into simpler—either in terms of graph structure, or problem dimension—mechanism-design problems by using threshold mechanisms as building blocks. We believe that these techniques will also find use in other mechanism-design problems.

By leveraging the decomposition method along with threshold mechanisms, we obtain various truthful, approximation mechanisms for Multi-VC, which yield the *first* truthful mechanisms for multidimensional vertex cover with non-trivial approximation guarantees. We obtain a truthful, $O(\log n)$-approximation mechanism (Theorem 4.8) for any proper minor-closed family of graphs (such as planar graphs). Our decomposition method shows that any instance of r-dimensional VC can be broken up into $O(r^2 \log n)$ instances of *single-dimensional VC*; this in turn leads to a truthful, $O(r^2 \log n)$-approximation mechanism for r-dimensional VC (Theorem 4.9). In particular, for any fixed r, we obtain an $O(\log n)$-approximation for any graph. Here n is the number of nodes.

It is worthwhile to note that in addition to their usefulness in the design of truthful, approximation mechanisms for Multi-VC, some of the mechanisms we design also enjoy good frugality properties. We obtain (Theorem 4.12) the *first* mechanisms for Multi-VC that are polytime, truthful and *simultaneously* achieve bounded approximation ratio *and* bounded frugality ratio with respect to the benchmarks in [5, 19]. This nicely complements a result of [5], who devise such a mechanism for single-dimensional VC.

Related work. As mentioned earlier, there is little prior work on the CM problem for multidimensional covering problems. Dughmi and Roughgarden [12] give a general technique to convert an FPTAS for an SWM problem to a truthful-in-expectation FPTAS. However, for covering problems, they obtain an additive approximation, which does not translate to a (worst-case) multiplicative approximation. In fact, as they observe, a multiplicative approximation ratio is impossible (in polytime) using their technique, or any other technique that constructs a MIDR mechanism whose range is a proper subset of all outcomes.

For single-dimensional covering problems, various other results, including black-box results, are known. Briest et al. [3] consider a closely-related generalization, which one may call the "single-value setting"; although this is a multidimensional setting, it admits a simple monotonicity condition sufficient for implementability, which makes this setting easier to deal with than our multidimensional settings. They show that a pseudopolynomial time algorithm (for covering and packing problems) can be converted into a truthful FPTAS. Lavi and Swamy [21] mainly consider packing problems, but mention that their technique also yields results for single-dimensional covering problems.

Single-dimensional covering problems have been well studied from the perspective of *frugality*. Here the goal is to design mechanisms that have bounded (over-)payment with respect to some benchmark, but one does not (typically) care about the cost of the solution returned. Starting with the work of Archer and Tardos [1], various benchmarks for frugality have been proposed and investigated for various problems including VC, k-edge-disjoint paths, spanning tree, s-t cut; see [18, 6, 19, 5] and the references therein. Some of our mechanisms for Multi-VC are inspired by the constructions in [19, 5], and simultaneously achieve bounded approximation ratio and bounded frugality ratio.

Our decomposition method, where we combine mechanisms for simpler problems into a mechanism for the given problem, is somewhat in the same spirit as the construction in [24]. They give a toolkit for combining truthful mechanisms, identifying sufficient conditions under which this combination preserves truthfulness. But they work only with the single-dimensional setting, which is much more tractable to deal with.

Finally, as noted earlier, there are a wide variety of results on truthful mechanism-design for packing SWM problems, such as combinatorial auctions [10, 21, 9, 13, 20, 14, 11].

2 Preliminaries

In a *multidimensional covering mechanism-design problem*, we have m items that need to be covered, and n agents/players who provide covering objects. Each agent i provides a set $\mathcal{T}_i$ of covering objects. All this information is public knowledge. We use $[k]$ to denote the set $\{1, \ldots, k\}$. Each agent i has a *private cost* (or type) vector $c_i = \{c_{i,v}\}_{v \in \mathcal{T}_i}$, where $c_{i,v}$ is the cost he incurs for providing object $v \in \mathcal{T}_i$; for $T \subseteq \mathcal{T}_i$, we use $c_i(T)$ to denote $\sum_{v \in T} c_{i,v}$. A feasible solution or allocation selects a subset $T_i \subseteq \mathcal{T}_i$ for each agent i, denoting that i provides the objects in T_i. Given this solution, each agent i incurs the private cost $c_i(T_i)$. Also, the mechanism designer incurs a publicly-known cost $pub(T_1, \ldots, T_n)$. The goal is to minimize the total cost $\sum_i c_i(T_i) + pub(T_1, \ldots, T_n)$ incurred. We call this the *cost minimization* (CM) problem. Note that we can encode any feasibility constraints in the covering problem by simply setting $pub(a) = \infty$ if a is not a feasible allocation. Observe that if we view the mechanism designer also as a player, then the CM problem is equivalent to maximizing the social welfare, which is given by $\sum_i -c_i(T_i) - pub(T_1, \ldots, T_n)$.

Various covering problems can be cast in the above framework. For example, in the mechanism-design version of *vertex cover* (Section 4), the items are edges of a graph. Each agent i provides a subset $\mathcal{T}_i$ of the nodes of the graph and incurs a private cost $c_{i,v}$ if node $v \in T_i$ is used to cover an edge. We can set $pub(T_1, \ldots, T_n) = 0$ if $\bigcup_i T_i$ is a vertex cover, and ∞ otherwise, to encode that the solution must be a vertex cover. It is also easy to see that the mechanism-design version of *uncapacitated facility location* (UFL; Section 3), where each agent provides some facilities and has private facility-opening costs, and the client-assignment costs are public, can be modeled by letting $pub(T_1, \ldots, T_n)$ be the total client-assignment cost given the set $\bigcup_i T_i$ of open facilities.

Let C_i denote the set of all possible cost functions of agent i, and $\mathcal{O}$ be the (finite) set of all possible allocations. Let $C = \prod_{i=1}^{n} C_i$. For a tuple $x = (x_1, \ldots, x_n)$, we use x_{-i} to denote $(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$. Similarly, let $C_{-i} = \prod_{j \neq i} C_j$. For an allocation $a = (T_1, \ldots, T_n)$, we sometimes use a_i to denote T_i, $c_i(a)$ to denote $c_i(a_i) = c_i(T_i)$. A (direct revelation) *mechanism* $M = (\mathcal{A}, p_1, \ldots, p_n)$ for a covering problem consists of an allocation algorithm $\mathcal{A} : C \mapsto \mathcal{O}$ and a payment function $p_i : C \mapsto \mathbb{R}$ for each agent i, and works as follows. Each agent i reports a cost function c_i (that might be different from his true cost function). The mechanism computes the allocation $\mathcal{A}(c) = (T_1, \ldots, T_n)$, and pays $p_i(c)$ to each agent i. Throughout, we use $\overline{c}_i$ to denote the true cost function of i. The *utility* $u_i(c_i, c_{-i}; \overline{c}_i)$ that player i derives when he reports c_i and the others report c_{-i} is $p_i(c) - \overline{c}_i(T_i)$, and each agent i aims to maximize his own utility (rather than the social welfare).

A desirable property for a mechanism to satisfy is *truthfulness*, wherein every agent i maximizes his utility by reporting his true cost function. All our mechanisms will also satisfy the natural property of *individual rationality* (IR), which means that every agent has nonnegative utility if he reports his true cost.

Definition 2.1 *A mechanism $M = \big(\mathcal{A}, \{p_i\}\big)$ is truthful if for every agent i, every $c_{-i} \in C_{-i}$, and every $\overline{c}_i, c_i \in C_i$, we have $u_i(\overline{c}_i, c_{-i}; \overline{c}_i) \geq u_i(c_i, c_{-i}; \overline{c}_i)$. M is IR if for every i, every $\overline{c}_i \in C_i$ and every $c_{-i} \in C_{-i}$, we have $u_i(\overline{c}_i, c_{-i}; \overline{c}_i) \geq 0$.*

To ensure that truthfulness and IR are compatible, we consider *monopoly-free* settings: for every player i, there is a feasible allocation a (i.e., $pub(a) < \infty$) with $a_i = \emptyset$. (Otherwise, if there is no such allocation, then i needs to be paid at least $\min_{v \in \mathcal{T}_i} c_{i,v}$ for IR, so he can lie and increase his utility arbitrarily.)

For a *randomized mechanism* M, where $\mathcal{A}$ or the p_i's are randomized, we say that M is *truthful in expectation* if each agent i maximizes his expected utility by reporting his true cost. We now say that M is IR if for *every* coin toss of the mechanism, the utility of each agent is nonnegative upon bidding truthfully.

Since the CM problem is often *NP*-hard, our goal is to design a mechanism $M = \big(\mathcal{A}, \{p_i\}\big)$ that is truthful (or truthful in expectation), and where $\mathcal{A}$ is a ρ-approximation algorithm; that is, for every input c, the solution $a = \mathcal{A}(c)$ satisfies $\sum_i c_i(a) + pub(a) \leq \rho \cdot \min_{b \in \mathcal{O}} \big(\sum_i c_i(b) + pub(b)\big)$. We call such a mechanism a *truthful, ρ-approximation mechanism.*

The following theorem gives a necessary and sometimes sufficient condition for when an algorithm $\mathcal{A}$ is *implementable*, that is, admits suitable payment functions $\{p_i\}$ such that $\big(\mathcal{A}, \{p_i\}\big)$ is a truthful mechanism. Say that $\mathcal{A}$ satisfies *weak monotonicity* (WMON) if for all i, all $c_i, c'_i \in C_i$, and all $c_{-i} \in C_{-i}$, if $\mathcal{A}(c_i, c_{-i}) = a$, $\mathcal{A}(c'_i, c_{-i}) = b$, then $c_i(a) - c_i(b) \leq c'_i(a) - c'_i(b)$. Define the dimension of a covering problem to be $\max_i |\mathcal{T}_i|$. It is easy to see that for a single-dimensional covering problem—so $C_i \subseteq \mathbb{R}$ for all i—WMON is equivalent to the following simpler condition: say that $\mathcal{A}$ is *monotone* if for all i, all $c_i, c'_i \in C_i$, $c_i \leq c'_i$, and all $c_{-i} \in C_{-i}$, if $\mathcal{A}(c_i, c_{-i}) = a$, $\mathcal{A}(c'_i, c_{-i}) = b$ then $b_i \subseteq a_i$.

Theorem 2.2 (Theorems 9.29 and 9.36 in [26]) *If a mechanism $\big(\mathcal{A}, \{p_i\}\big)$ is truthful, then $\mathcal{A}$ satisfies WMON. Conversely, if the problem is single-dimensional, or if C_i is convex for all i, then every WMON algorithm $\mathcal{A}$ is implementable.*

3 A black-box reduction for multidimensional metric UFL

In this section, we consider the multidimensional metric *uncapacitated facility location* (UFL) problem and present a *black-box reduction* from truthful mechanism design to algorithm design. We show that any ρ-approximation algorithm for UFL satisfying an additional property can be converted in a black-box fashion to a truthful-in-expectation ρ-approximation mechanism (Theorem 3.1). This is the first such result for a multidimensional covering problem. As a corollary, we obtain a truthful-in-expectation, 2-approximation mechanism (Corollary 3.3).

In the mechanism-design version of UFL, we have a set $\mathcal{D}$ of clients that need to be serviced by facilities, and a set $\mathcal{F}$ of locations where facilities may be opened. Each agent i may provide facilities at the locations in $\mathcal{T}_i \subseteq \mathcal{F}$. By making multiple copies of a location if necessary, we may assume that the $\mathcal{T}_i$s are disjoint. Hence, we will simply say "facility ℓ" to refer to the facility at location $\ell \in \mathcal{F}$. For each facility $\ell \in \mathcal{T}_i$ that is opened, i incurs a private opening cost of $\overline{f}_{i,\ell}$, and assigning client j to an open facility ℓ incurs a publicly known assignment/connection cost $c_{\ell j}$. To simplify notation, given a tuple $\{f_{i,\ell}\}_{i\in[n],\ell\in\mathcal{T}_i}$ of facility costs, we use f_ℓ to denote $f_{i,\ell}$ for $\ell \in \mathcal{T}_i$. The goal is to open a subset $F \subseteq \mathcal{F}$ of facilities, so as to minimize $\sum_{\ell\in F}\overline{f}_\ell + \sum_{j\in\mathcal{D}}\min_{\ell\in F} c_{\ell j}$. We will assume throughout that the $c_{\ell j}$s form a metric. It will be notationally convenient to allow our algorithms to have the flexibility of choosing the open facility $\sigma(j)$ to which a client j is assigned (instead of $\text{argmin}_{\ell\in F} c_{\ell j}$); since assignment costs are public, this does not affect truthfulness, and any approximation guarantee achieved also clearly holds when we drop this flexibility.

We can formulate (metric) UFL as an integer program, and relax the integrality constraints to obtain the following LP. Throughout, we use ℓ to index facilities in $\mathcal{F}$ and j to index clients in $\mathcal{D}$.

$$\min \quad \sum_\ell f_\ell y_\ell + \sum_{j,\ell} c_{\ell j} x_{\ell j} \qquad \text{s.t.} \qquad \sum_\ell x_{\ell j} \geq 1 \quad \forall j, \qquad 0 \leq x_{\ell j} \leq y_\ell \leq 1 \quad \forall \ell, j. \qquad \text{(FL-P)}$$

Here, $\{f_\ell\}_\ell = \{f_{i,\ell}\}_{i\in[n],\ell\in\mathcal{T}_i}$ is the vector of reported facility costs. Variable y_ℓ denotes if facility ℓ is opened, and $x_{\ell j}$ denotes if client j is assigned to facility ℓ; the constraints encode that each client is assigned to a facility, and that this facility must be open.

Say that an algorithm $\mathcal{A}$ is a *Lagrangian multiplier preserving* (LMP) ρ-approximation algorithm for UFL if for every instance, it returns a solution $\big(F, \{\sigma(j)\}_{j\in\mathcal{D}}\big)$ such that $\rho\sum_{\ell\in F} f_\ell + \sum_j c_{\sigma(j)j} \leq \rho \cdot OPT_{\text{(FL-P)}}$. The main result of this section is the following black-box reduction.

Theorem 3.1 *Given a polytime, LMP ρ-approximation algorithm $\mathcal{A}$ for UFL, one can construct a polytime, truthful-in-expectation, individually rational, ρ-approximation mechanism M for multidimensional UFL.*

Proof : We build upon the convex-decomposition idea used in [21]. The randomized mechanism M works as follows. Let $f = \{f_\ell\}$ be the vector of reported facility-opening costs, and c be the public connection-cost metric.

1. Compute the optimal solution (y^*, x^*) to (FL-P) (for the input (f, c)). Let $\{p_i^* = p_i^*(f)\}$ be the payments made by the *fractional* VCG mechanism that outputs the optimal LP solution for every input. That is, $p_i^* = \big(\sum_\ell f_\ell y'_\ell + \sum_{\ell,j} c_{\ell j} x'_{\ell j}\big) - \big(\sum_{\ell\notin\mathcal{T}_i} f_\ell y^*_\ell + \sum_{\ell,j} c_{\ell j} x^*_{\ell j}\big)$, where (y', x') is the optimal solution to (FL-P) with the additional constraints $y_\ell = 0$ for all $\ell \in \mathcal{T}_i$.

2. Let $\mathbb{Z}(P) = \{(y^{(q)}, x^{(q)})\}_{q\in\mathcal{I}}$ be the set of all integral solutions to (FL-P). In Lemma 3.2, we prove the key technical result that using $\mathcal{A}$, one can compute, in polynomial time, nonnegative multipliers $\{\lambda^{(q)}\}_{q\in\mathcal{I}}$ such that $\sum_q \lambda^{(q)} = 1$, $\sum_q \lambda^{(q)} y_\ell^{(q)} = y^*_\ell$ for all ℓ, and $\sum_{q,\ell,j} \lambda^{(q)} c_{\ell j} x_{\ell j}^{(q)} \leq \rho\sum_{\ell,j} c_{\ell j} x^*_{\ell j}$.

3. With probability $\lambda^{(q)}$: (a) output the solution $\big(y^{(q)}, x^{(q)}\big)$; (b) pay $p_i^{(q)}$ to agent i, where $p_i^{(q)} = 0$ if $\sum_{\ell\in\mathcal{T}_i} f_\ell y^*_\ell = 0$, and $\sum_{\ell\in\mathcal{T}_i} f_\ell y_\ell^{(q)} \cdot \frac{p_i^*}{\sum_{\ell\in\mathcal{T}_i} f_\ell y^*_\ell}$ otherwise.

Clearly, M runs in polynomial time. Fix a player i. Let $\overline{f}_i$ and f_i be the true and reported cost vector of i. Let f_{-i} be the reported cost vectors of the other players. Let (y^*, x^*) be an optimal solution to (FL-P) for (f, c). Note that $\mathrm{E}\big[p_i(f)\big] = p_i^*(f)$ If $\sum_{\ell\in\mathcal{T}_i} f_\ell y^*_\ell = 0$ then this follows since $p_i^*(f) = 0$ (because then (y^*, x^*) is also an optimal solution to (FL-P) when player i does not participate). Otherwise, this follows since $\sum_q \lambda^{(q)} y^{(q)} = y^*_\ell$ for all ℓ. So $\mathrm{E}\big[u_i(f_i, f_i; \overline{f}_i)\big] = \mathrm{E}\big[p_i\big] - \sum_q \lambda^{(q)} \sum_{\ell\in\mathcal{T}_i} \overline{f}_\ell y_\ell^{(q)} = p_i^*(f) - \sum_{\ell\in\mathcal{T}_i} \overline{f}_\ell y^*_\ell$ where the last equality is again because $\sum_q \lambda^{(q)} y^{(q)} = y^*_\ell$ for all ℓ. Since p_i^* and y^* are respectively the payment to i and the assignment computed for input (f_i, f_{-i}) by the fractional VCG mechanism, which is truthful, it follows that player i maximizes his utility in the VCG mechanism, and hence, his expected utility under mechanism M, by reporting his true opening costs. Thus, M is truthful in expectation.

This also implies the ρ-approximation guarantee because the convex decomposition obtained in Step 2 shows that the expected cost of the solution computed by M for input (f, c) (where we may assume that f is the true cost vector) is at most $\rho \cdot OPT_{(\text{FL-P})}(f, c)$. Finally, since the fractional VCG mechanism is IR, for any agent i, the VCG payment $p_i^*(f)$ satisfies $p_i^*(f) \geq \sum_{\ell \in \mathcal{T}_i} f_\ell y_\ell^*$, and therefore $p_i^{(q)} \geq \sum_{\ell \in \mathcal{T}_i} f_\ell y_\ell^{(q)}$. So M is IR. ■

Lemma 3.2 *The convex decomposition in step 2 can be computed in polytime.*

Proof : It suffices to show that the LP (P) can be solved in polynomial time and its optimal value is 1. Recall that $\{(y^{(q)}, x^{(q)})\}_{q \in \mathcal{I}}$ is the set of all integral solutions to (FL-P).

$$\begin{array}{llr} \max & \sum_q \lambda^{(q)} & \text{(P)} \\ \text{s.t.} & \sum_q \lambda^{(q)} y_\ell^{(q)} = y_\ell^* \qquad \forall \ell & (1) \\ & \sum_{j,\ell,q} \lambda^{(q)} c_{\ell j} x_{\ell j}^{(q)} \leq \rho \sum_{j,\ell} c_{\ell j} x_{\ell j}^* & (2) \\ & \sum_q \lambda^{(q)} \leq 1 & (3) \\ & \lambda \geq 0. & \end{array}$$

$$\begin{array}{llr} \min & \sum_\ell y_\ell^* \alpha_\ell + \big(\rho \sum_{j,\ell} c_{\ell j} x_{\ell j}^*\big)\beta + z & \text{(D)} \\ \text{s.t.} & \sum_\ell y_\ell^{(q)} \alpha_\ell + \big(\sum_{j,\ell} c_{\ell j} x_{\ell j}^{(q)}\big)\beta + z \geq 1 \qquad \forall q & (4) \\ & z, \beta \geq 0. & \end{array}$$

Since (P) has an exponential number of variables, we consider the dual (D). Here the α_ℓs, β and z are the dual variables corresponding to constraints (1), (2), and (3) respectively. Clearly, $OPT_{(\text{D})} \leq 1$ since $z = 1$, $\alpha_\ell = 0 = \beta$ for all ℓ is a feasible dual solution. If there is a feasible dual solution (α', β', z') of value smaller than 1, then the rough idea is that by running $\mathcal{A}$ on the UFL instance with facility costs $\{\frac{\alpha'_\ell}{\rho}\}$ and connection costs $\{\beta' c_{\ell j}\}$, we can obtain an integral solution whose constraint (4) is violated. (This idea needs be modified a bit since α'_ℓ could be negative; see below.) Hence, we can solve (D) efficiently via the ellipsoid method using $\mathcal{A}$ to provide the separation oracle. This also yields an equivalent dual LP consisting of only the polynomially many violated inequalities found during the ellipsoid method. The dual of this compact LP gives an LP equivalent to (P) with polynomially many $\lambda^{(q)}$ variables whose solution yields the desired convex decomposition.

We now fill in the details. Suppose (α', β', z') is feasible to (D) and $\sum_\ell y_\ell^* \alpha'_\ell + (\rho \sum_{j,\ell} c_{\ell j} x_{\ell j}^*)\beta' + z' < 1$. Define $a^+ := \max(0, a)$; for a vector $v = (v_1, \ldots, v_n)$, define $v^+ := (v_1^+, \ldots, v_n^+)$. Consider the UFL instance with facility costs $\{f'_\ell = \alpha'^+_\ell / \rho\}$ and connection costs $\{c'_{\ell j} = \beta' c_{\ell j}\}$. (Clearly c' is also a metric.) Running $\mathcal{A}$ on this input, we can obtain an integral solution $(y^{(q)}, x^{(q)})$ such that

$$\rho \sum_\ell \frac{\alpha'^+_\ell}{\rho} y_\ell^{(q)} + \sum_{j,\ell} \beta' c_{\ell j} x_{\ell j}^{(q)} \leq \rho \cdot OPT_{(\text{FL-P})}(f', c') \leq \rho \Big(\sum_\ell \frac{\alpha'^+_\ell}{\rho} y_\ell^* + \sum_{j,\ell} \beta' c_{\ell j} x_{\ell j}^* \Big).$$

Clearly the facilities ℓ with $\alpha'_\ell \leq 0$ contribute 0 to the LHS and RHS of the above inequality. Now consider the integer solution $\hat{y}^{(q)}$ where $\hat{y}_\ell^{(q)}$ is 1 if $\alpha'_\ell \leq 0$ and is y_ℓ^* otherwise. Adding $\sum_{\ell : \alpha'_\ell \leq 0} \alpha'_\ell \hat{y}_\ell^{(q)}$ to the LHS and $\sum_{\ell : \alpha'_\ell \leq 0} \alpha'_\ell y_\ell^*$ to the RHS of the above inequality, since $y_\ell^* \leq 1$ for all ℓ and $\alpha'^+_\ell = \alpha'_\ell$ when $\alpha'_\ell > 0$, we infer that

$$\sum_\ell \alpha'_\ell \hat{y}_\ell^{(q)} + \sum_{j,\ell} \beta' c_{\ell j} x_{\ell j}^{(q)} \leq \sum_\ell \alpha'_\ell y_\ell^* + \big(\rho \sum_{j,\ell} c_{\ell j} x_{\ell j}^*\big)\beta' < 1 - z'$$

which contradicts that (α', β', z') is feasible to (D). Hence, $OPT_{(\mathrm{D})} = OPT_{(\mathrm{P})} = 1$.

Thus, we can add the constraint $\sum_\ell y^*_\ell \alpha_\ell + (\rho \sum_{j,\ell} c_{\ell j} x^*_{\ell j})\beta + z \leq 1$ to (D) without altering anything. If we solve the resulting LP using the ellipsoid method, and take the inequalities corresponding to the violated inequalities (4) found by $\mathcal{A}$ during the ellipsoid method, then we obtain a compact LP with only a polynomial number of constraints that is equivalent to (D). The dual of this compact LP yields an LP equivalent to (P) with a polynomial number of $\lambda^{(q)}$ variables which we can solve to obtain the desired convex decomposition. ■

By using the polytime LMP 2-approximation algorithm for UFL devised by Jain et al. [17], we obtain the following corollary of Theorem 3.1.

Theorem 3.3 *There is a polytime, IR, truthful-in-expectation, 2-approximation mechanism for multidimensional UFL.*

4 Truthful mechanisms for multidimensional VC

We now consider the multidimensional vertex-cover problem (VC), and devise various polytime, truthful, approximation mechanisms for it. We often use Multi-VC to distinguish multidimensional VC from its algorithmic counterpart.

Recall that in Multi-VC, we have a graph $G = (V, E)$ with n nodes. Each agent i provides a subset $\mathcal{T}_i$ of nodes. For simplicity, we first assume that the $\mathcal{T}_i$s are disjoint, and given a cost-vector $\{c_{i,u}\}_{i \in [n], u \in \mathcal{T}_i}$, we use c_u to denote $c_{i,u}$ for $u \in \mathcal{T}_i$. Notice that monopoly-free then means that each $\mathcal{T}_i$ is an independent set. In Remark 4.6 we argue that many of the results obtained in this disjoint-$\mathcal{T}_i$s setting (in particular, Theorems 4.8 and 4.9) also hold when the $\mathcal{T}_i$s are not disjoint (but each $\mathcal{T}_i$ is still an independent set). The goal is to choose a minimum-cost *vertex cover*, i.e., a min-cost set $S \subseteq V$ such that every edge is incident to a node in S.

As mentioned earlier, VC becomes a rather challenging mechanism-design problem in the *multidimensional* mechanism-design setting. Whereas for *single-dimensional VC*, many of the known 2-approximation algorithms for VC are implementable, none of these underlying techniques yield implementable algorithms even for the simplest multidimensional setting, 2-dimensional VC, where *every player owns at most two nodes*; see Appendix B and C for examples. Moreover, no maximal-in-distributional-range (MIDR) mechanism whose range is a proper subset of all outcomes can achieve a bounded multiplicative approximation guarantee [12].[1] This also rules out the convex-decomposition technique of [21], which yields MIDR mechanisms.

We develop two main techniques for Multi-VC in this section. In Section 4.1, we introduce a simple class of truthful mechanisms called *threshold mechanisms*, and show that although seemingly restricted, threshold mechanisms can achieve non-trivial approximation guarantees. In Section 4.2, we develop a *decomposition method* for Multi-VC that uses threshold mechanisms as building blocks and gives a general way of reducing the mechanism-design problem for Multi-VC into simpler mechanism-design problems.

By leveraging the decomposition method along with threshold mechanisms, we obtain various truthful, approximation mechanisms for Multi-VC, which yield the *first* truthful mechanisms for multidimensional vertex cover with non-trivial approximation guarantees. (1) We obtain a truthful, $O(\log n)$-approximation mechanism (Theorem 4.8) for any proper minor-closed family of graphs (such as planar graphs). (2) We show that any instance of r-dimensional VC can be decomposed

[1] If $\mathcal{A}$ is a randomized MIDR algorithm and S is an inclusion-wise minimal vertex cover such that the range of $\mathcal{A}$ does not include a distribution that returns S with probability 1, then $\mathcal{A}$ incurs non-zero cost on the instance where the cost of a node u is 0 if $u \in S$ and is 1 (say) otherwise, and so its approximation ratio is unbounded.

into $O(r^2 \log n)$ *single-dimensional* VC instances; this leads to a truthful, $O(r^2 \log n)$-approximation mechanism for r-dimensional VC (Theorem 4.9). In particular, for any fixed r, we obtain an $O(\log n)$-approximation.

Theorem 4.12 shows that our mechanisms also enjoy good frugality properties. We obtain the first mechanisms for Multi-VC that are polytime, truthful, and achieve bounded approximation ratio *and* bounded frugality ratio. This nicely complements a result of [5], who devise such mechanisms for single-dimensional VC.

4.1 Threshold Mechanisms

Definition 4.1 *A* threshold mechanism *M for Multi-VC works as follows. On input c, for every i and every node $u \in \mathcal{T}_i$, M computes a threshold $t_u = t_u(c_{-i})$ (i.e., t_u does not depend on i's reported costs). M then returns the solution $S = \{v \in V : c_v \leq t_v\}$ as the output, and pays $p_i = \sum_{u \in S \cap \mathcal{T}_i} t_u$ to agent i.*

If t_u only depends on the costs in the neighbor-set $N(u)$ of u, for all $u \in V$ (note that $N(u) \cap \mathcal{T}_i = \emptyset$ if $u \in \mathcal{T}_i$), we call M a *neighbor-threshold mechanism.* A special case of a neighbor-threshold mechanism is an *edge-threshold mechanism*: for every edge $uv \in E$ we have edge thresholds $t_u^{(uv)} = t_u^{(uv)}(c_v)$, $t_v^{(uv)} = t_v^{(uv)}(c_u)$, and the threshold of a node u is given by $t_u = \max_{v \in N(u)}(t_u^{(uv)})$.

In general, threshold mechanisms may not output a vertex cover, however it is easy to argue that threshold mechanisms are always truthful and IR.

Lemma 4.2 *Every threshold mechanism for Multi-VC is IR and truthful.*

Proof : IR is immediate from the definition of payments. To see truthfulness, fix an agent i. For every $\overline{c}_i, c_i \in C_i, c_{-i} \in C_{-i}$ we have $u_i(c_i, c_{-i}; \overline{c}_i) = \sum_{v \in \mathcal{T}_i : c_v \leq t_v}(t_v - \overline{c}_v)$. It follows that i's utility is maximized by reporting $c_i = \overline{c}_i$. ■

Inspired by [19, 5], we define an *x-scaled* edge-threshold mechanism as follows: fix a vector $(x_u)_{u \in V}$, where $x_u > 0$ for all u, and set $t_u^{(uv)} := x_u c_v / x_v$ for every edge (u, v). We abuse notation and use $\mathcal{A}_x$ to denote both the resulting edge-threshold mechanism and its allocation algorithm. Also, define $\mathcal{B}_x$ to be the neighbor-threshold mechanism where we set $t_u := \sum_{v \in N(u)} x_u c_v / x_v$. Define $\alpha(G; x) := \max_{u \in V}\big(\max_{S \subseteq N(u) : S \text{ independent}} \frac{x(S)}{x_u}\big)$.

Lemma 4.3 *$\mathcal{A}_x$ and $\mathcal{B}_x$ output feasible solutions and have a tight approximation ratio $\alpha(G; x) + 1$.*

Proof : Clearly, every node selected by $\mathcal{A}_x$ is also selected by $\mathcal{B}_x$. So it suffices to show that $\mathcal{A}_x$ is feasible, and to show the approximation ratio for $\mathcal{B}_x$. For any edge (u, v), either $c_u \leq x_u c_v / x_v$ and u is output, or $c_v \leq x_v c_u / x_u$ and v is output. So $\mathcal{A}_x$ returns a vertex cover.

Let S be the output of $\mathcal{B}_x$ on input c, and let S^* be a min-cost vertex cover. We have $c(S) = c(S \cap S^*) + c(S \setminus S^*) \leq c(S^*) + \sum_{u \in S \setminus S^*} t_u = c(S^*) + \sum_{u \in S \setminus S^*} \sum_{v \in N(u)} x_u c_v / x_v$. Note that $S \setminus S^*$ is an independent set since S^* is a vertex cover, so $\sum_{u \in S \setminus S^*} \sum_{v \in N(u)} x_u c_v / x_v \leq \sum_{v \in S^*} \frac{c_v}{x_v} \sum_{u \in N(v) capS^*} x_u \leq \sum_{v \in S^*} c_v \cdot \alpha(G; x)$. Hence $c(S) \leq (\alpha(G; x) + 1)c(S^*)$. The tightness of the approximation guarantee follows from Example 1 below. ■

Corollary 4.4 *(i) Setting $x = \vec{1}$ gives $\alpha(G; x) \leq \Delta(G)$, which is the maximum degree of a node in G, so $\mathcal{A}_{\vec{1}}$ has approximation ratio at most $\Delta(G) + 1$.*
(ii) Taking x to be the eigenvector corresponding to the largest eigenvalue $\lambda_{\max}$ of the adjacency matrix of G ($x > 0$ by the Perron-Frobenius theorem) gives $\alpha(G; x) \leq \lambda_{max}$ (see [5]), so $\mathcal{A}_x$ has approximation ratio $\lambda_{max} + 1$.

Example 1 (Tightness of approximation ratio of $\mathcal{A}_x$ and $\mathcal{B}_x$) Let u and $S \subseteq N(u)$ achieve the maximum in the definition of $\alpha(G;x)$. Now consider the instance (G,c) where $c_u = x_u$, $c_v = x_v$ for all $v \in S$ and $c_w = 0$ for all $w \in V \setminus (\{u\} \cup S)$. The mechanism $\mathcal{A}_x$ will choose $\{u\} \cup S$ in the output, whereas $V \setminus S$ is a vertex cover of cost $c_u = x_u$. So, $\mathcal{A}_x$ has approximation ratio at least $\frac{x_u + x(S)}{x_u} = 1 + \alpha(G;x)$.

Although neighbor-threshold mechanisms are more general than edge-threshold mechanisms, Lemma 4.5 (proved in Appendix A) shows that this yields limited dividends in the approximation ratio. Define $\alpha'(G) = \min_{\text{orientations of } G}\bigl(\max_{u \in V, S \subseteq N^{\text{in}}(u): S \text{ independent}} |S|\bigr)$, where $N^{\text{in}}(u) = \{v \in N(u) : (u,v) \text{ is directed into } u\}$. Note that $\alpha'(G) \leq \alpha(G;\vec{1}) \leq \Delta(G)$. If $G = (V,E)$ is *everywhere γ-sparse*, i.e., $|\{(u,v) \in E : u,v \in S\}| \leq \gamma|S|$ for all $S \subseteq V$, then $\alpha'(G) \leq \gamma$; this follows from Hakimi's theorem [16]. A well-known result in graph theory states that for every proper family $\mathcal{G}$ of graphs that is closed under taking minors (e.g., planar graphs), there is a constant γ, such that every $G \in \mathcal{G}$ is has at most $\gamma|V(G)|$ edges [23] (see also [8], Chapter 7, Exer. 20); since $\mathcal{G}$ is minor-closed, this also implies that G is *everywhere* γ-sparse, and hence $\alpha'(G) \leq \gamma$ for all $G \in \mathcal{G}$.

Lemma 4.5 *A (feasible) neighbor-threshold mechanism M for graph G with approximation ratio ρ, yields an $O\bigl(\rho\log(\alpha'(G))\bigr)$-approximation edge-threshold mechanism for G. This implies an approximation ratio of (i) $O(\rho\log\gamma)$ if G is an everywhere γ-sparse graph; (ii) $O(\rho)$ if G belongs to a proper minor-closed family of graphs (where the constant in the $O(.)$ depends on the graph family).*

Remark 4.6 Any neighbor-threshold mechanism M with approximation ratio ρ that works under the disjoint-$\mathcal{T}_i$s assumption can be modified to yield a truthful, ρ-approximation mechanism when we drop this assumption. Let $A_u = \{i : u \in \mathcal{T}_i\}$. Set $\hat{c}_u = \min_{i \in A_u} c_{i,u}$ for each $u \in V$ and let $\hat{t}_u$ be the neighbor-threshold of u for the input $\hat{c}$. Note that $\hat{t}_u$ depends only on c_{-i} for every $i \in A_u$. Set $t^i_u := \min\{\hat{t}_u, \min_{j \neq i: u \in \mathcal{T}_j} c_{j,u}\}$ for all $i, u \in \mathcal{T}_i$. Consider the threshold mechanism M' with $\{t^i_u\}$ thresholds, where we use a fixed tie-breaking rule to ensure that we pick u for at most one agent $i \in A_u$ with $c_{i,u} = t^i_u$. Then the outputs of M on c, and of M' on input $\hat{c}$ coincide. Thus, M' is a truthful, ρ-approximation mechanism.

4.2 A decomposition method

We now propose a general reduction method for Multi-VC that uses threshold mechanisms as building blocks to reduce the task of designing truthful mechanisms for Multi-VC to the task of designing threshold mechanisms for simpler (in terms of graph structure or the dimensionality of the problem) Multi-VC problems. This reduction is useful because designing good threshold mechanisms appears to be a much more tractable task for Multi-VC. By utilizing the threshold mechanisms designed in Section 4.1 in our decomposition method, we obtain an $O(\log n)$-approximation mechanism for any proper minor-closed family of graphs, and an $O(r^2 \log n)$-approximation mechanism for r-dimensional VC.

A *decomposition mechanism* M for $G = (V,E)$ is constructed as follows.

- Let $G_1, \ldots, G_k$ be subgraphs of G such that $\bigcup_{q=1}^k E(G_q) = E$,
- Let $M_1, \ldots, M_k$ be threshold mechanisms for $G_1, \ldots, G_k$ respectively. For any $v \in V$, let t^q_v be v's threshold in M_q if $v \in V(G_i)$, and 0 otherwise.
- Define M to be the threshold mechanism obtained by setting the threshold for each node v to $t_v := \max_{q=1,\ldots,k}(t^q_v)$ for any $v \in V$. The payments of M are then as specified in Definition 4.1. Notice that if all the M_is are neighbor threshold mechanisms, then so is M.

Lemma 4.7 *The decomposition mechanism M described above is IR and truthful. If $\rho_1, \ldots, \rho_k$ are the approximation ratios of $M_1, \ldots, M_k$ respectively, then M has approximation ratio $\left(\sum_q \rho_q\right)$.*

Proof : Since M is a threshold mechanism, it is IR and truthful by Lemma 4.2. The optimal vertex cover for G induces a vertex cover for each subgraph G_q. So M_q outputs a vertex cover S_q of cost at most $\rho_q \cdot OPT$, where OPT is the optimal vertex-cover cost for G. It is clear that M outputs $\bigcup_q S_q$, which has cost at most $\left(\sum_q \rho_q\right) \cdot OPT$. ■

Theorem 4.8 *If $G = (V, E)$ is everywhere γ-sparse, then one can devise a polytime, $O(\gamma \log |V|)$-approximation decomposition mechanism for G. Hence, there is a polytime, truthful, $O(\log n)$-approximation mechanism for* Multi-VC *on any proper minor-closed family of graphs. These guarantees also hold when the $\mathcal{T}_i$s are not disjoint.*

Proof : Let $n = |V|$. Since $|E| \leq \gamma n$, there are at most $n/2$ nodes with degree larger than 4γ. Let H_1 be the subgraph of G consisting of the edges incident to the vertices of G with degree at most 4γ. Now, $G_1 = G \setminus H_1$ (i.e., we delete the nodes and edges of H_1 to obtain G_1) is also γ-sparse. So, we can similarly find a subgraph H_2 that contains at least half of the nodes of G_1. Continuing this process, we obtain subgraphs $H_1, \ldots, H_k$ that partition G, where each subgraph H_q has maximum degree at most 4γ and $|V(H_q)| \geq |V(G \setminus (H_1 \cup \ldots H_{q-1})|/2$. Hence, $k \leq \log n$. Using the (edge-threshold) mechanism $\mathcal{A}_{\vec{1}}$ defined in Corollary 4.4, for each subgraph gives a $(4\gamma+1)$-approximation for each H_q, and hence a $(4\gamma + 1) \log n$-approximation neighbor-threshold mechanism for G. By Remark 4.6, this also holds when the $\mathcal{T}_i$s are not disjoint.

As noted in Section 4.1, every proper minor-closed family of graphs is everywhere γ-sparse for some $\gamma > 0$. Thus, the above result implies a truthful, $O(\log n)$-approximation for any proper minor-closed family (where the constant in the $O(.)$ depends on the graph family; e.g., for planar graphs $\gamma \leq 4$). ■

Complementing Theorem 4.8, we next present another decomposition mechanism whose guarantee depends only on the dimensionality of the problem, and not on the underlying graph structure.

Theorem 4.9 *For any r-dimensional instance of* Multi-VC *on $G = (V, E)$, one can obtain a polytime, $O(r^2 \log |V|)$-approximation, decomposition mechanism, even when the $\mathcal{T}_i$s are not disjoint.*

Proof : We decompose G into single-dimensional subgraphs, by which we mean subgraphs that contain at most one node from each $\mathcal{T}_i$. Initialize $j = 1$, $V_j = \emptyset$. While, $\bigcup_{q=1}^{j-1} E(G_q) \neq E$, we do the following: for every agent i, we pick one of the nodes of $\mathcal{T}_i$ uniformly at random and add it to V_j. We also add all the nodes in $V \setminus \left(\bigcup_{i=1}^{n} \mathcal{T}_i\right)$ to V_j. Let G_j be the induced subgraph on V_j; set $j \leftarrow j + 1$.

For any edge $e \in E$, the probability that both of its ends appear in some subgraph G_j, for any $i = 1, \ldots, l$, is at least $1/r^2$. So, the expected value of $|E \setminus \bigcup_{q=1}^{j-1} E(G_q)|$ decreases by a factor of at least $(1 - 1/r^2)$ with j. Hence, the expected number of subgraphs produced above is $O\left(\frac{\log |E|}{\log(r^2/(r^2-1))}\right) = O(r^2 \log |V|)$ (this also holds with high probability). Each G_j yields a single-dimensional VC instance (where a node may be owned by multiple players). Any truthful mechanism for a 1D-problem is a threshold mechanism. So we can use any truthful, 2-approximation mechanism for single-dimensional VC for the G_js and obtain an $O(r^2 \log n)$-approximation for r-dimensional Multi-VC. ■

The following lemma shows that the decomposition obtained above into single-dimensional subgraphs is essentially the best that can hope for, for $r = 2$.

Lemma 4.10 *There are instances of* 2*-dimensional VCP that require* $\Omega(\log |V(G)|)$ *single-dimensional subgraphs in any decomposition of* G.

Proof : Define G^n to be the bipartite graph with vertices $\{u_1, \dots, u_n, v_1, \dots, v_n\}$ and edges $\{(u_i, v_j) : i \neq j\}$. Each agent $i = 1, \dots, n$ owns vertices u_i and v_i.

For $n = 2$ the claim is obvious. Let q_n be the minimum number of single-dimensional subgraphs needed to decompose G^n. Suppose the claim is true for all $j < n$ and we have decomposed G^n into single-dimensional subgraphs $D = \{G_1, \dots, G_{q_n}\}$. We may assume that $V(G_1) = \{u_1, \dots, u_k, v_{k+1}, \dots, v_n\}$ (if G_1 has less than n nodes, pad it with extra nodes). Let H_1 and H_2 be the subgraphs of G induced by $\{u_1, \dots, u_k, v_1, \dots, v_k\}$ and $\{u_{k+1}, \dots, u_n, v_{k+1}, \dots, v_n\}$, respectively. The graphs in $D \setminus \{G_1\}$ must contain a decomposition of H_1 and a decomposition of H_2. So $q_n \geq 1 + \max(q_k, q_{n-k})$, and hence, by induction, we obtain that $q_n \geq 1 + (1 + \log_2(n/2)) = 1 + \log_2 n$.
■

Frugality considerations. Karlin et al. [18] and Elkind et al. [6] propose various benchmarks for measuring the *frugality ratio* of a mechanism, which is a measure of the (over-)payment of a mechanism. The mechanisms that we devise above also enjoy good frugality ratios with respect to the following benchmark introduced by [6], which is denoted by $\nu(G, c)$ in [19] (and $\text{NTU}_{\max}$ in [6]).

Definition 4.11 (Frugality benchmark $\boldsymbol{\nu(G, c)}$ [18, 6]) *Given an instance of VC on a graph* $G = (V, E)$ *with node costs* $\{c_u\}$*, we define* $\nu(G, c)$ *as follows. Fix an arbitrary min-cost vertex cover* S *(with respect to* c*).*[2]

$$\begin{aligned} \nu(G,c) \ := \ \max \quad & \sum_{v \in S} x_v \\ \textit{s.t.} \quad & x_v \geq c_v && \textit{for all } v \in S \\ & \sum_{v \in S \setminus T} x_v \leq \sum_{v \in T \setminus S} c_v && \textit{for all vertex covers } T. \end{aligned}$$

The *frugality ratio* of a mechanism $M = (\mathcal{A}, \{p_i\})$ on G is defined as $\phi_M(G) := \sup_c \frac{\sum_i p_i(c)}{\nu(G,c)}$. The proof of Lemma 4.3 is easily modified to show that the x-scaled mechanism $\mathcal{A}_x$ satisfies $\sum_i p_i(c) \leq \sum_u t_u \leq \beta(G; x) c(V)$, where $\beta(G; x) = \max_{u \in V} \frac{x(N(u))}{x_u}$. Since [6] show that $\nu(G, c) \geq c(V)/2$, this implies that $\phi_{\mathcal{A}_x}(G) \leq 2\beta(G; x)$. Also, if M is a decomposition mechanism constructed from threshold mechanisms $M_1, \dots, M_k$, where each M_q satisfies $\sum_u t_u^q \leq \phi_q \cdot c(V(G_q))$, then it is easy to see that $\phi_M(G) \leq 2 \sum_{q=1}^k \phi_q$. Thus, we obtain the following results.

Theorem 4.12 *Let* $G = (V, E)$ *be a graph with* n *nodes. We can obtain a polytime, truthful, IR mechanism* M *with the following approximation* $\rho = \rho_M(G)$ *and frugality* $\phi = \phi_M(G)$ *ratios.*

(i) $\rho = (\beta(G; x) + 1)$, $\phi \leq 2\beta(G; x)$ *for Multi-VC on* G*;*

(ii) $\rho, \phi = O(\gamma \log n)$ *for Multi-VC on* G *when* G *is everywhere* γ*-sparse; hence, we achieve* $\rho, \phi = O(\log n)$ *for Multi-VC on any proper minor-closed family;*

(iii) $\rho = O(r^2 \log n)$, $\phi = O(r^2 \log n \cdot \Delta(G))$ *for* r*-dimensional Multi-VC on* G *(using a* 2*-approximation mechanism with frugality ratio* $2\Delta(G)$ *[6] for single-dimensional VC in the construction of Theorem 4.9).*

[2] Elkind et al. [6] prove that $\nu(G, c)$ does not depend on the specific min-cost vertex cover S used in the definition.

References

[1] A. Archer and É. Tardos. Frugal path mechanisms. *ACM TALG*, 3(1), 2007.

[2] I. Ashlagi, S. Dobzinski, and R. Lavi. An optimal lower bound for anonymous scheduling mechanisms. In *Proceedings of the 10th EC*, pages 169–176, 2009.

[3] P. Briest, P. Krysta, and B. Vöcking. Approximation techniques for utilitarian mechanism design. In *Proceedings of the 37th STOC*, pages 39–48, 2005.

[4] E. Clarke. Multipart pricing of public goods. *Public Choice*, 11:17–33, 1971.

[5] N. Chen, E. Elkind, N. Gravin, and F. Petrov. Frugal mechanism design via spectral techniques. In *Proceedings of the 51st FOCS*, pages 755–764, 2010.

[6] E. Elkind, L. Goldberg, and P. Goldberg. Frugality ratios and improved truthful mechanisms for vertex cover. In *Proceedings of the 8th EC*, pages 336–345, 2007.

[7] N. Devanur, M. Mihail, and V. Vazirani. Strategyproof cost-sharing mechanisms for set cover and facility location games. *Decision Supp. Syst.*, 39(1):11–22, 2005.

[8] Reinhard Diestel. *Graph Theory*, Springer, August 2005.

[9] S. Dobzinski. Two randomized mechanisms for combinatorial auctions. In *Proceedings of the 11th APPROX*, pages 89–103, 2008.

[10] S. Dobzinski, N. Nisan, and M. Schapira. Truthful randomized mechanisms for combinatorial auctions. *J. Comput. Syst. Sci*, 78(1):15–25, 2012.

[11] S. Dobzinski and J. Vondrák. The computational complexity of truthfulness in combinatorial auctions. In *Proceedings of the 13th EC*, pages 405–422, 2012.

[12] S. Dughmi and T. Roughgarden. Black-box randomized reductions in algorithmic mechanism design. In *Proceedings of the 51st FOCS*, pages 775–784, 2010.

[13] S. Dughmi, T. Roughgarden, and Q. Yan. From convex optimization to randomized mechanisms: toward optimal combinatorial auctions. In *STOC*, 149–158, 2011.

[14] S. Dughmi and J. Vondrák. Limitations of Randomized Mechanisms for Combinatorial Auctions. In *Proceedings of the 52nd FOCS*, 502–511, 2011.

[15] T. Groves. Incentives in teams. *Econometrica*, 41(4):617–31, July 1973.

[16] S. Hakimi. On the degrees of the vertices of a directed graph. *J. Franklin Inst.*, 279:290–308, 1965.

[17] K. Jain, M. Mahdian, E. Markakis, A. Saberi, and V. Vazirani. Greedy facility location algorithms analyzed using dual fitting with factor-revealing LP. *JACM*, 50:795–824, 2003.

[18] A. Karlin, D. Kempe, T. Tamir. Beyond VCG: Frugality of Truthful Mechanisms In *Proceedings of the 46th FOCS*, pages 615–626, 2005.

[19] D. Kempe, M. Salek, and C. Moore. Frugal and truthful auctions for vertex covers, flows and cuts. In *Proceedings of the 51st FOCS*, pages 745–754, 2010.

[20] R. Lavi, A. Mu'alem, and N. Nisan. Towards a characterization of truthful combinatorial auctions. In *Proceedings of the 44th FOCS*, pages 574–583, 2003.

[21] R. Lavi and C. Swamy. Truthful and near-optimal mechanism design via linear programming. *Journal of the ACM*, 58(6): 25, 2011

[22] R. Lavi and C. Swamy. Truthful mechanism design for multidimensional scheduling via cycle monotonicity. *Games and Economic Behavior*, 67(1):99–124, 2009.

[23] W. Mader. Homomorphieeigenschaften und mittlere kantendichte von graphen. *Mathematische Annalen*, 174:265–268, 1967.

[24] A. Mu'alem and N. Nisan. Truthful approximation mechanisms for restricted combinatorial auctions. *Games and Economic Behavior*, 64(2):612–631, 2008.

[25] N. Nisan and A. Ronen. Algorithmic mechanism design. *Games and Economic Behavior*, 35:166–196, 2001.

[26] N. Nisan, T. Roughgarden, É. Tardos, and V. Vazirani. *Algorithmic Game Theory.* Cambridge University Press, September 2007.

[27] W. Vickrey. Counterspeculation, auctions and competitive sealed tenders. *Journal of Finance*, 16:8–37, 1961.

A Proof of Lemma 4.5

Statements (i) and (ii) follow from the statement for general graphs and the graph-theoretic facts mentioned before Lemma 4.5, so we focus on proving the statement for an arbitrary graph G. Let $\alpha' = \alpha'(G)$.

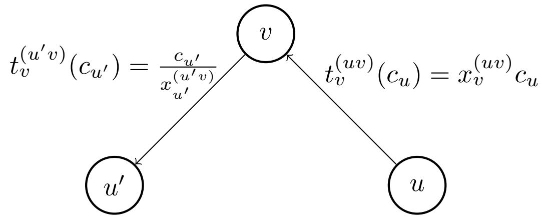
Consider an arbitrary vertex $v \in V$. For any $u \in N(v)$ define $x_v^{(uv)} := \inf\{\sigma \geq 0 : t_u(c_v = \beta, c_{-v} = \vec{0}) \geq 1 \;\; \forall \beta \geq \sigma\}$.

Claim 1: $x_v^{(uv)} < \infty$. If not, then for any $p > 0$, there exists $q \geq p$ such that $t_u(c_v = q, c_{-v} = \vec{0}) < 1$. So, let $p = \rho$ and $q \geq p$ be such that $t_u(c_v = q, c_{-v} = \vec{0}) < 1$. Consider the cost vector c where $c_u = 1$, $c_v = q$, and $c_z = 0$ for $z \neq u, v$, we see that the approximation ratio ρ is contradicted for the instance (G, c) (i.e., graph G with the cost vector c): $V \setminus v$ is an optimal vertex cover of cost 1 but the threshold mechanism does not choose u so it chooses v as it is feasible and incurs cost $q > \rho$.

Claim 2: $x_v^{(uv)} > 0$. If $x_v^{(uv)} = 0$, then similar to the above, by considering c where $c_u = 1$, $c_v = \epsilon$, $c_z = 0$ for $z \neq u, v$, where ϵ is very small, we see that M outputs u, which means M does not have the approximation ratio ρ.

Now orient the edges of G according to the orientation that determines $\alpha'(G)$ to obtain the directed graph D. For any arc (u, v) in D, consider linear edge-threshold functions $t_v^{(uv)}(c_u) = x_v^{(uv)} c_u$, and $t_u^{(uv)}(c_v) = (1/x_v^{(uv)})c_v$. Using these edge-thresholds we obtain an edge-threshold mechanism M'. M' is feasible since for any arc (u, v) if u is not chosen by M', we should have $c_u > t_u^{(uv)}(c_v) = (1/x_v^{(uv)})c_v$ which implies $t_v^{(uv)}(c_u) = x_v^{(uv)} c_u > c_v$ hence v is chosen by M'.

We assert that M' has approximation ratio $O(\rho \log(\alpha'))$. Note that if T is the outcome of M'


v
$t_v^{(u'v)}(c_{u'}) = \frac{c_{u'}}{x_{u'}^{(u'v)}}$
$t_v^{(uv)}(c_u) = x_v^{(uv)} c_u$
u'
u

and T^* is the optimal outcome, then we have

$$\begin{aligned} c(T) &= c(T\cap T^*) + c(T\setminus T^*) \le c(T^*) + \sum_{w\in T\setminus T^*} \max_{u\in N(w)} t_w^{(uw)}(c_u)\\ &\le c(T^*) + \sum_{w\in T\setminus T^*}\sum_{u\in N(w)} t_w^{(uw)}(c_u) = c(T^*) + \sum_{\substack{w\in T\setminus T^*\\ u\in N(w)}} c_u t_w^{(uw)}(1)\\ &= c(T^*) + \sum_{u\in T^*} c_u \sum_{w\in N(u)\cap(T\setminus T^*)} t_w^{(uw)}(1) \qquad (\text{since } N(w)\subseteq T^* \text{ for } w\notin T^*) \end{aligned}$$

Note that $T \setminus T^*$ is an independent set, so it suffices to show for any $u \in V(G)$, if $S \subseteq N(u)$ forms an independent set then $\sum_{w\in S} t_w^{(uw)}(1) \le \rho(\log(\alpha') + 2)$.

Let $\delta^{out}(u) = \{v : (u,v) \in D\}$, $S_1 := S \cap \delta^{out}(u)$, and $S_2 := S \setminus S_1$. So, we have

$$\sum_{w\in S} t_w^{(uw)}(1) = \sum_{w\in S_1} t_w^{(uw)}(1) + \sum_{w\in S_2} t_w^{(uw)}(1) = \sum_{w\in S_1} x_w^{(uw)} + \sum_{w\in S_2} \tfrac{1}{x_u^{(uw)}} \tag{5}$$

Choose an arbitrary $w \in S_1$. By definition of $x_w^{(uw)}$, for every $\epsilon_w \ge 0$, there is some $0 \le \delta_w \le \epsilon_w$ such that $t_u(c_w = x_w^{(uw)} - \epsilon_w + \delta_w, \vec{0}) < 1$. Hence, $u \notin M(G, \hat{c})$ where $\hat{c}_w = x_w^{(uw)} - \epsilon_w + \delta_w$, $\hat{c}_u = 1$, and $\hat{c}_z = 0$ otherwise. So, since $M(G, \hat{c})$ is a vertex cover, we should have $w \in M(G, \hat{c})$ which means $t_w(c_u = 1, \vec{0}) \ge x_w^{(uw)} - \epsilon_w + \delta_w$. Thus, as S_1 is an independent set, for the cost vector c' where $c'_u = 1$, $c'_w = x_w^{(uw)} - \epsilon_w + \delta_w$ if $w \in S_1$, and $c'_z = 0$ otherwise, we have $S_1 \subseteq M(G, c')$ (since $t_w(c'_{N(w)}) = t_w(c_u = 1, \vec{0})$). Letting ϵ_w tend to 0, we get that $\rho \ge \sum_{w\in S_1} x_w^{(uw)}$, as $V \setminus N(u)$ is a vertex cover of cost 1.

Let $S_2 = \{v_1, \ldots, v_k\}$ where $x_u^{(uv_1)} \le x_u^{(uv_2)} \le \ldots \le x_u^{(uv_k)}$. Consider c'' where $c''_u = x_u^{(uv_q)}$, $c''_z = 1$ if $z \in S_2$, and $c''_z = 0$ otherwise. Then, $\{v_1, \ldots, v_q\} \subseteq M(G, c'')$ hence $\rho \ge q/x_u^{(uv_l)}$ for each $q = 1, \ldots, k$. So, $\sum_{q=1}^{k} \frac{1}{x_u^{(uv_q)}} \le \sum_{q=1}^{k} \rho/q \le \rho(\log(|S_2|) + 1) \le \rho\log(\alpha') + \rho$. Therefore, (5) gives

$$\sum_{w\in S} t_w^{(uw)}(1) \le \rho + \rho\log(\alpha') + \rho = \rho(\log(\alpha') + 2). \qquad \blacksquare$$

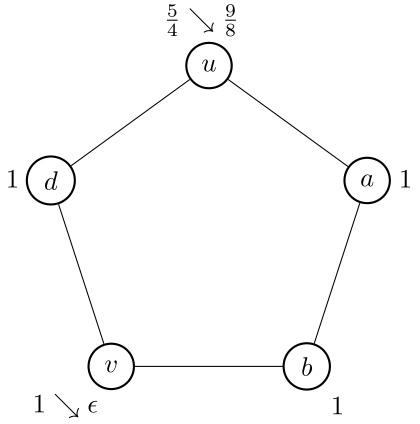
B LP-rounding does not work for Multi-VC

A common method for designing approximation algorithms for VC (and in general) is to solve the following LP-relaxation and then round the optimal solution.

$$\min \quad \sum_v c_v x_v \qquad \text{s.t.} \qquad x_u + x_v \ge 1 \quad \forall (u,v) \in E. \tag{VC-P}$$

We show that *any* LP-rounding algorithm that always includes nodes with $x_u \ge \frac{1}{2}$ and does not include any node u with $x_u = 0$ is not WMON.

Example 2 Consider the graph G shown below where u and v belong to agent 1. For the cost-vector $(c_u, c_a, c_b, c_v, c_d) = (5/4, 1, 1, 1, 1)$, the unique optimal solution to the LP is $(x_u, x_a, x_b, x_v, x_d) = (1/2, 1/2, 1/2, 1/2, 1/2)$. Therefore, the algorithm includes both u and v in the output.



G

Consider the cost vector $c' = (c'_1, c_{-1})$ where agent 1 reduces the costs for u and v to $c'_u = 9/8$ and $c'_v = \epsilon < 1/16$ (all other costs are unchanged). Then WMON dictates that both u and v must still be chosen. However, the unique optimal solution to the LP with the new costs is $x_a = x_d = x_v = 1,\ x_u = x_b = 0$ with cost $2 + \epsilon$. (This follows because if $x_u = 1$ then the cost of an LP solution is at least $1 + 9/8$; if $x_u = 1/2$, then the cost of an LP solution is at least $9/16 + 1 + 1/2$; both are greater than $2 + \epsilon$ as $\epsilon < 1/16$.) So M will not output u, which contradicts WMON.

The above example also shows that the following well-known combinatorial 2-approximation algorithm for VC does not satisfy WMON: Given a graph $G = (V, E)$, construct a bipartite graph G' having two copies of V, say V_1, V_2, and having edges $(u_1, v_2), (u_2, v_1)$ for every edge $(u, v) \in E$; solve VC on G' and if any of the copies of a node are chosen in this solution, then pick that node in the solution for G.

In the above example, for the cost-vector c, every optimal vertex cover for G' includes exactly one copy of u and one copy of v, so both u and v will be chosen in the solution for G. For the cost-vector c', no optimal vertex cover for G' includes any copies of u, so u will not be chosen in the solution for G. This contradicts WMON.

C Primal-dual methods do not work for Multi-VC

The dual of (VC-P) is as follows.

$$\max \quad \sum_e y_e \qquad \text{s.t.} \qquad \sum_{e \in \delta(v)} y_e \le c_v \quad \forall v \in V. \tag{VC-D}$$

Various primal-dual algorithm based on dual ascent are known to yield 2-approximation algorithms. All of these start with $y = \vec{0}$, raise dual variables while maintaining dual feasibility, and return the nodes whose costs are completely "paid" by the dual variables.

The two most common variants are where one fixes an ordering of the edges in which to raise dual variables, and where one raises all (unfrozen) dual variables simultaneously. We show that neither of these lead to WMON algorithms.

Example 3 Consider the graph shown in Fig. 1, where the dual variables are increased in the order ux, xy, yv, and u and v belong to one agent.

Let $c_u = 1,\ c_x = 1.5,\ c_y = 1.05,\ c_v = 0.5$. The primal-dual algorithm will output $\{u, x, v\}$. Now, if we reduce c_u to 0.5 and c_v to 0.3, and keep c_x and c_y unchanged, the algorithm outputs $\{u, x, y\}$ which contradicts WMON.

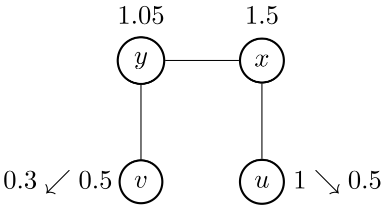


Figure 1:

Example 4 Now consider the simultaneous-dual-ascent primal-dual algorithm. Consider again the same graph as in Example 3 but with a different assignment of costs, as shown in Fig. 2. Let $c_u = 1,\ c_x = 3,\ c_y = 4.6,\ c_v = 2.5$. The primal-dual algorithm outputs $\{u, x, v\}$. Now, if we reduce c_u to 0.5 and c_v to 2.4 and keep c_x and c_y unchanged, the algorithm outputs $\{u, y\}$, which contradicts WMON.

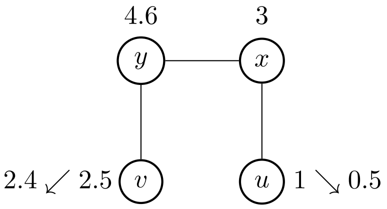


Figure 2: